

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
S1	19	thread near2 switch same flag	USPAT	OR	ON	2006/07/18 13:06
S2	22	non-blocking near2 synchronization	USPAT	OR	ON	2006/07/17 16:06
S3	19	718/107,108.ccls. and non-blocking	USPAT	OR	ON	2006/07/17 16:12
S4	0	718/107,108.ccls. and frontier adj pointer	USPAT	OR	ON	2006/07/17 16:12
S5	0	"718".clas. and frontier adj pointer	USPAT	OR	ON	2006/07/17 16:21
S6	19	"718".clas. and thread same non-blocking	USPAT	OR	ON	2006/07/17 16:19
S7	3	"718".clas. and thread same context adj switching and non-blocking	USPAT	OR	ON	2006/07/17 16:19
S8	12	thread same context adj switching and non-blocking	USPAT	OR	ON	2006/07/17 16:19
S9	1	"707".clas. and frontier adj pointer	USPAT	OR	ON	2006/07/17 16:22
S10	22	non-blocking near2 synchronization	USPAT	OR	ON	2006/07/17 16:22
S11	26	non-blocking near2 synchroniz\$	USPAT	OR	ON	2006/07/17 16:22
S12	4	S11 not S10	USPAT	OR	ON	2006/07/17 16:25
S13	50	thread near2 synchroniz\$ and context adj switching	USPAT	OR	ON	2006/07/17 18:02
S14	5	non adj blocking near2 synchroniz\$ same thread	USPAT	OR	ON	2006/07/17 18:10
S15	328	718/107,108.ccls. and thread	USPAT	OR	ON	2006/07/17 18:11
S16	202	718/107,108.ccls. and thread.ab.	USPAT	OR	ON	2006/07/17 18:11
S17	21	718/107,108.ccls. and thread.ab. same synchroniz\$	USPAT	OR	ON	2006/07/17 18:18
S18	3	718/107,108.ccls. and non-blocking same synchroniz\$	USPAT	OR	ON	2006/07/17 18:19
S19	12	"718".clas. and non-blocking same synchroniz\$	USPAT	OR	ON	2006/07/17 18:30
S20	12	"718".clas. and non adj blocking same synchroniz\$	USPAT	OR	ON	2006/07/17 18:32
S21	80	718/107,108.ccls. and thread near5 synchroniz\$	USPAT	OR	ON	2006/07/17 18:32
S22	35	718/107,108.ccls. and thread near5 synchroniz\$ and repeat\$	USPAT	OR	ON	2006/07/17 18:51
S23	1	"5094004".pn.	USPAT	OR	ON	2006/07/17 18:52
S24	332	"718".clas. and interrupt\$ same thread	USPAT	OR	ON	2006/07/17 18:53
S25	37	"718".clas. and interrupt\$ same thread.ab.	USPAT	OR	ON	2006/07/17 18:53

EAST Search History

S26	1	("6370596").PN.	USPAT	OR	OFF	2006/07/18 18:32
S27	1	("5463643").PN.	USPAT	OR	OFF	2006/07/18 18:32

[Next](#) [Up](#) [Previous](#)

Next: [About this document ...](#) **Up:** [Improving Wait-Free Algorithms for](#) **Previous:** [Acknowledgments](#)

Bibliography

- 1 J. Alemany and W. Felten.
Performance issues in non-blocking synchronization on shared-memory multiprocessors.
In *Proceedings of the 11th ACM Symposium on Principles of Distributed Computing*, pages 125-134, 1992.
- 2 J. Anderson, S. Ramamurthy, and K. Jeffay.
Real-time computing with lock-free shared objects.
In *Proceedings of the 16th IEEE Real-Time Systems Symposium*, pages 28-37, Dec 1995.
- 3 T. P. Baker.
Stack-based scheduling of realtime processes.
RTSYSTS: Real-Time Systems, 3, 1991.
- 4 Greg Barnes.
A method for implementing lock-free data structures.
In *5th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA 93)*, 1993.
- 5 B. Bershad.
Practical considerations for non-blocking concurrent objects.
In *IEEE International Conference on Distributed Computing Systems*, pages 264-273, 1993.
- 6 James E. Burns and Gary L. Peterson.
Constructing multi-reader atomic values from non-atomic values.
In *ACM Symposium on Principles of Distributed Computing*, pages 222-231, 1987.
- 7 J. Chen and A. Burns.
A fully asynchronous reader/writer mechanism for multiprocessor real-time systems.
Technical Report YCS-288, Department of Computer Science, University of York, 1997.
- 8 J. Chen and A. Burns.
A three-slot asynchronous reader/writer mechanism for multiprocessor real-time systems.
Technical Report YCS-186, Department of Computer Science, University of York, 1997.
- 9 P. Courtois, F. Heymans, and D. Parnas.
Concurrent control with readers and writers.
Communications of the Association of Computing Machinery, 14(10):667-668, 1971.
- 10 E. Dijkstra.
Solution of a problem in concurrent programming control.
Communications of the Association of Computing Machinery, 8(9):569, 1965.
- 11 J. Eliot and B. Moss.
Lock-free garbage collection for multiprocessors.
In *Parallel Algorithms and Architectures*, 1991.

- 12 M. Greenwald and D. Cheriton.
The synergy between non-blocking synchronization and operating system structure.
In *Operating Systems Design and Implementation*, pages 123-136, 1996.
- 13 M. Herlihy.
A methodology for implementing highly concurrent data structure.
Proceeding of the 2nd ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, pages 197-206, 1989.
- 14 M. Herlihy.
Wait-free synchronization.
ACM Transactions on Programming Languages and Systems, 13(1):124-149, January 1991.
- 15 M. Herlihy.
A methodology for implementing highly concurrent data objects.
ACM Transactions on Programming Languages and Systems, 15(5):745-770, November 1993.
- 16 H. Kopetz and *et al.*
Distributed fault-tolerant real-time systems: the Mars approach.
IEEE Micro, 9(1):25-40, 1989.
- 17 H. Kopetz and J. Reisinger.
The non-blocking write protocol NBW: A solution to a real-time synchronisation problem.
In *IEEE Real-Time Systems Symposium*, pages 131-137, 1993.
- 18 C. M. Krishna and K. G. Shin.
Real-Time Systems.
McGraw-Hill, 1997.
- 19 Anthony LaMarca.
A performance evaluation of lock-free synchronization protocols.
In *ACM Symposium on Principles of Distributed Computing*, pages 130-140, 1994.
- 20 L. Lamport.
Concurrent reading and writing.
Communications of ACM, 20(11):806-811, Nov 1977.
- 21 C. L. Liu and James W. Layland.
Scheduling algorithms for multiprogramming in a hard-real-time environment.
Journal of the ACM, 20(1):46-61, 1973.
- 22 H. Massalin and C. Pu.
A lock-free multiprocessor os kernel.
Technical Report CUCS-005-91, Department of Computer Science, Columbia University, 1991.
- 23 M. Moir.
Practical implementations of non-blocking synchronization primitives.
In *ACM Symposium on Principles of Distributed Computing*, pages 219-228, 1997.
- 24 G. Peterson.

- Concurrent reading while writing.
ACM Transactions on Programming Languages and Systems, 5(1):46-55, 1983.
- 25 S. Poledna, T. Mocken, and J. Schiemann.
Ercos: an operating system for automotive applications.
Technical Report 960623, Society of Automotive Engineers Technical Paper Series, 1996.
- 26 S. Prakash, Y.-H. Lee, and T. Johnson.
Non-blocking algorithms for concurrent data structures.
Technical Report 91-002, Department of Computer Science, University of California, Los Angeles, 1991.
- 27 S. Prakash, Y.-H. Lee, and T. Johnson.
A non-blocking algorithm for shared queues using compare-and-swap.
IEEE Transactions on Computers, 43(5):548-559, 1994.
- 28 R. Rajkumar.
Synchronization in real-time systems - a priority inheritance approach.
Kluwer Academic Publishers, 1991.
- 29 L. Sha, R. Rajkumar, and J. P. Lehoczky.
Priority inheritance protocols: An approach to real-time synchronization.
IEEE Transactions on Computers, 39(9):1175-1185, 1990.
- 30 Nir Shavit and Dan Touitou.
Software transactional memory.
In *ACM Symposium on Principles of Distributed Computing*, pages 204-213, 1995.
- 31 H. R. Simpson.
Four-slot fully asynchronous communication mechanism.
In *IEEE Proceedings*, 1990.
- 32 John Turek, Dennis Shasha, and Sundeep Prakash.
Locking without blocking: making lock based concurrent data structure algorithms nonblocking.
In *ACM Proceedings of the Principles of Database Systems*, pages 212-222, 1992.
- 33 J. D. Valois.
Implementing lock-free queues.
In *Proceedings of the Seventh International Conference on Parallel and Distributed Computing Systems*, pages 64-69, Las Vegas, NV, 1994.
- 34 K. Vidyasankar.
Concurrent reading while writing revisited.
Distributed Computing, 4(2):81-86, 1990.
- 35 K. Zuberi, P. Pillai, and K. G. Shin.
EMERALDS: a small-memory real-time microkernel.
In *ACM Symposium on Operating Systems Principles*, volume 34, pages 277-299, 1999.

hai huang 2002-03-30